

REMARKS

In view of the foregoing amendments and following remarks responsive to the Office Action of June 18, 2003, Applicant respectfully requests favorable reconsideration of this Application.

112 Rejections

In Sections 3 and 4 of the Office Action, the Office rejected claims 27 and 29 because they did not make sense. The Office's point is well taken and Applicant has herein rewritten these claims to correct a clerical error that caused these claims to be printed incorrectly in the previous amendment. Applicant has also slightly amended claims 26 and 28, from which claims 27 and 29 depended, to further improve their form.

In Section 5 of the Office Action, the Office rejected claims 27 and 29 because their preambles incorrectly identified them as method claims. Applicant has herein rewritten these claims to correct this error.

Prior Art Rejections

In Sections 6-9 of the Office Action, the Office rejected all of claims 1-29 in view of prior art. Specifically, the Office rejected claims 1-8 and 10-29 as anticipated by Mishra and rejected claim 9 as obvious over Mishra in view of Rosen. The rejections are, in large part, duplicates of the rejections set forth in the previous Office Action. The minor differences include the rejection of new claims 26-29, which were not pending when the previous Office Action was issued, and minimal changes in some of

the column and line number references within the Mishra reference for a few of the claimed features.

The Office also added a new section entitled ***Response to Arguments*** addressing the arguments against the prior art rejections that Applicant offered in response to the previous Office Action.

The Present Invention

The present invention is a method and apparatus for automatically opening files of particular types on a computer using attributes such as window size and window position dictated by how the user previously positioned and sized windows when viewing files of the same type. It further includes the concept of, when a user opens a certain file of a first type (the first file), automatically opening a second file that has some file name attribute relative to the file name of the first file. For instance, whenever a file having a particular given name with a first file type extension, e.g., johnsmith.doc, is opened, the computer will automatically open a second file having the same file name but a file name extension of a second type, e.g., johnsmith.pdf. The invention is particularly useful for users who repeatedly open one or more files of certain types that they would like to be sized and positioned in the same place every time and/or repeatedly need to open two related files and view them simultaneously, such as might be necessary for repetitive data entry tasks.

In its broadest aspect as recited in independent Claims 1 and 21, the software of the present invention remembers at least one display attribute of a file being used by a

user, for instance, the position and size of the window in which the file is displayed. Then, when the user opens another file of the same file type, it will automatically open in a window in the same position and of the same size as a previous file of the same type. The invention can be applied to several different file types so that a user can open multiple files that he/she may need to view simultaneously and they will always open up in the position and size windows that the user desires.

The Mishra Reference

Mishra discloses a method for displaying plurality of image sets simultaneously on a computer screen. More specifically, the user can tailor the display protocol to each type of workstation he/she frequents. This yields a descriptive triple viewing: session type/for user/on workstation type. There is no discussion in Mishra of file types or of remembering the size, position or any other attribute of a window associated with a particular file type.

While Mishra appears to disclose the concept of making the individual windows in a work place user adjustable, it accomplishes this in an entirely different way than the present invention that does not involve any dependence on file type determination, let alone storing the user settings when a last file of a particular file type was closed and using those settings for subsequent files of the same file type.

Mishra speaks of user customizable image viewing sessions in which "image sets" are displayed in user defined workspaces, each workspace having a user defined position and size. For instance, a user may define a session type for reviewing and

editing a book. That session type may include three workspaces, including a first workspace for displaying chapters, a second workspace for displaying the table of contents, and a third workspace for displaying the index. In the terminology of Mishra, the chapters of the book are a first “image set”, the index is a second “image set” and the table of contents is a third “image set”.

Traversal of Prior Art Rejections

In view of the present Office Action, it now appears that Applicant may have been misinterpreting the Office’s position with respect to Mishra in its response to the previous Office Action. In view of the present Office Action, it appears that the Office is equating the “image sets” of Mishra with the files of a certain file type in the present application and is taking the position that, when a plurality of images in an “image set” of Mishra are sequentially displayed in a user-defined workspace of a particular viewing session, this meets the key limitations of the claims of the present invention.

Applicant respectfully traverses.

There is virtually no discussion in the specification of Mishra as to how an “image set” is defined. In fact, there is nothing in Mishra suggesting that an “image set” comprises more than one file. Accordingly, without even considering any further issues, Mishra cannot anticipate any claim of the present invention since all of the claims recite multiple files of the same type being displayed in a window with the same attributes. Obviously, if Mishra’s “image set” is a single file, it cannot possibly meet the claim limitations.

Any assumption about the nature of Mishra's "image sets" that is not logically inherent from what is expressly disclosed in Mishra would simply be impermissible. Certainly, it is not inherent that an image set comprises more than one file. For instance, a multi-page .pdf file (i.e., Adobe Acrobat) or .tif file is a single file comprising multiple images.

However, even if one assumed, as the Office appears to be doing, that Mishra's "image set" comprises multiple files, it still would not anticipate the present invention because there is nothing in Mishra that discloses associating and storing a window attribute as a function of a file type and then applying that window attribute to subsequent files of the same file type when they are opened. Particularly, despite the lack of disclosure in Mishra as to the nature of an "image set", it is at least clear that an "image set" is a predefined set of images. Thus, no matter how the images that make up an "image set" are defined in Mishra, one thing that is clear is that an "image set" comprises a predefined set of images. Thus, even assuming that an image set comprises multiple files and that the files are of the same type, Mishra opens subsequent images with the same workspace attributes because the subsequent file is in the predefined "image set" corresponding to that workspace, not because the software determined the file type of the new file and then looked up the stored window attribute corresponding to that file type.

In this regard, Mishra is essentially just like any other windows-based software. For instance, when a user opens a windows-based application, e.g., a web browser,

and customizes its position and/or size, every file (e.g., web page) that the user opens subsequently will open in that window and thus have the same position and size attributes as the last opened file. Frequently, the files will be of the same type, e.g., HTML files.

However, this is merely a similar result to the present invention reached by an entirely different mechanism. The web browser will open in the window any file that it is capable of opening. While the result is similar, the manner in which it is reached is completely different and does not comprise the steps of “(2) storing data indicating a value of at least one attribute of the manner in which said first file was displayed associated with the data indicating a type of said first file” or “(3) when a next file of the type of said first file is opened by an operator for display, accessing said stored data indicating said value of said at least one attribute; and (4) displaying said next file using said value of said at least one attribute” as recited in claim 1. (Note that Applicant has amended independent claims 1 and 21 to emphasize the process of reaching the result, rather than the result: particularly, the claims now recite accessing the stored data and displaying the files in accordance with the stored data rather than simply displaying the data in accordance with the stored data).

Instead, Mishra (and many other windows-based applications) perform the completely different steps of (a) storing data indicating a display attribute of a window (that may have been customized by the user) and (b) opening all files in the corresponding “image set” in that window.

In other words, what is wholly lacking from Mishra is a stored association between a file type and a window attribute that is applied to subsequently opened files based on their file type. Like the present invention, Mishra concerns displaying multiple windows simultaneously, each window for displaying a different type of data, i.e., a different image set. However, these similarities do not overcome the basic failure of Mishra to disclose storing an association of a file type with a display window attribute and then applying that attribute when opening a new file of the particular file type.

Accordingly, claim 1 clearly distinguishes over Mishra.

Claim 21 is an independent apparatus claim containing essentially substantively identical imitations as claim 1. Accordingly, claim 21 distinguishes over the prior art for all of the same reasons as claim 1.

Furthermore, all other claims depend from one of claims 1 and 21, and, therefore, distinguish over prior art for all of the same reasons given above with respect to claims 1 and 21.

In addition, at least dependent claims 2, 3, 12, and 25 recite additional limitations that even further distinguish over the prior art of record. For instance, claim 2 depends from claim 1 and adds that the step of storing the window attribute in association with the file type comprises "storing said value when said first file is closed by said operator". This limitation relates to the fact that the operation of the present invention is transparent to the user. Particularly, the present invention simply automatically saves the window attribute in association with the file type whenever a file of that type is

closed. This is completely different than Mishra. In Mishra, there is a complicated process for defining attributes of the workspace. Furthermore, and more significantly, that process is performed entirely independently of the files that are displayed in the workspace. Specifically, in Mishra, the user defines a viewing session using the Display Protocol Specification algorithm. Referring to column 19, line 17 through column 21, line 28 of Mishra, the user uses the Display Protocol Specification algorithm to define the workspaces that make up a Viewing Session. This is done entirely without reference of any particular file that is to be displayed in a given workspace.

Accordingly, the limitations of claim 2 are not found in Mishra.

Even further, claim 3 depends from claim 2 and adds that the stored value "is the value of said attribute when said first file was closed". Again, Mishra cannot possibly teach this since there are no files involved when defining the workspace attribute.

Furthermore, claim 12 depends from claim 1 and recites a secondary feature of the present invention, namely, automatically opening a related second file of a second file type responsive to the opening of a file of a first file type. An example of this was given above in this amendment and in the specification, in which the relationship between the files was that they had the same name portion, but different extensions.

There is nothing in Mishra suggesting this feature. Once again, the result in Mishra is somewhat similar to the result of the present invention, but it is reached by a different process. Specifically, in a given viewing session in Mishra, such as the book editing viewing session mentioned above, there are three workspaces that display three different related "image sets", i.e., chapters of the book, table of contents of the book, and

index of the book. However, once again, Mishra achieves a similar result by an entirely different mechanism.

Applicant certainly does not dispute that Mishra discloses displaying two files of two different file types simultaneously and that those files are in two different user-defined windows. However, claim 12 does not recite merely such features. It recites “when a next file of the type of said first file is open for display, automatically opening a next file of the same type as said second file and having the same relationship to said next file as second file as to said first file.” With respect to this limitation, the Office cited the rejection of claim 1, step 3. However, this does not make sense. Claim 1, step 3 relates to opening a subsequent file of the same type, not simultaneously opening a second file of a different type.

Accordingly, this rejection is clearly unsupportable and should be withdrawn.

Dependent claim 25 depends from independent apparatus claim 21 and is substantively similar to claim 12. Accordingly, it distinguishes over the prior art of record for the same reasons given with respect to Claim 12.

Dependent claims 26 and 27 which depend from claim 12 and dependent claims 28 and 29, which depend from claim 25 recite specific examples of the feature described in Claims 12 and 25 pertaining to automatically opening a second file of a second type when a first file is opened. Particularly, claims 26 and 28 define that the “relationship” between the first and second files is that they have file names with identical portions. Claims 27 and 29 depend from claims 26 and 28, respectively, and

further add that the identical portions are the file names (minus the file name extensions).

For the reasons discussed above in connection with claim 12, Mishra obviously does not teach anything like this. Particularly, Mishra does not teach anything relating to using file names and file name extensions to determine how and when to open a file.

Illustrative Example Comparing Mishra to the Present Invention

The discussion above of the claim elements that are not met by Mishra sufficiently set forth the reasons why the present invention as claimed is patentably distinct over Mishra. However, the following exercise should put to rest any possible lingering doubts as to how significant these distinctions are. Let us compare using Mishra's device versus the present invention in the specific example discussed in the background section of the present application. Particularly, the specification describes the following situation involving a data entry clerk for a financial institution, such as a bank.

One aspect of this individual's responsibilities includes transferring information from one computer file to another. For example, let us consider a bank in which information about its customers had previously been stored in signature cards which have now been scanned and stored as image files in a computer database. Such information might include the customer's name, address, social security number, mother's maiden name, and a copy of the customer's signature. As part of a scheme to automate customer banking, the bank has scanned all of its signature cards into a computer and converted them into .gif format (Graphic Interface File) files. Each signature card has been converted into two image files representing the front and the back of each card, respectively. The front image file is named #name.gif and the back image file is named #name.back.gif, wherein "#name" represents the customer's actual name.

It is the data entry clerk's responsibility to open up each of these .gif files and manually type in data from those signature cards into a database using a database application program. Accordingly, for each customer, the data entry clerk (1) opens the .gif file corresponding to the front of the signature card, (2) opens the file corresponding to the back of the same signature card, (3) sizes the windows displaying the two files so as to take up the top left hand quadrant of the screen and the bottom left hand quadrant of the screen, respectively, (4) opens up another window which provides a GUI (Graphical User Interface) in which the data entry clerk can type in information from the two aforementioned .gif files, (5) sizes and positions that window to take up the right-hand half of the screen, and (6) enters the data shown in the signature card files displayed in the left hand half of his computer screen into the GUI which is on the right hand half of the screen. These steps must be repeated for every customer. (Page 1, line 13 – page 3, line 2 of specification).

In a preferred embodiment of the present invention, the clerk opens the .gif file and sizes and positions it in the position desired, then opens the .back.gif file and sizes and positions it in the position desired and then opens the third file, in which the clerk will enter the data from the .gif and .back.gif files and sizes and positions it where desired. The clerk then transcribes the data from the first two files into the third file and then closes all three files. In accordance with the present invention, when the clerk opens the next .gif file, it will automatically open with the exact same position and size as the previous one when it was closed and will also automatically open up the .back.gif file that has the same name portion.

The clerk also could use Mishra's invention to achieve greater efficiency in this task compared to the prior art situation discussed in the background section of the present application. However, this would be done in an entirely different manner. Particularly, the clerk would first invoke the Display Specification Protocol algorithm of


Mishra and define the three desired workspaces entirely independently of any particular files. The clerk would then have to define for each of those workspaces an "image set". The image set for the first window, e.g., the window that needs to display the .gif files, would comprise all of the .gif files. As noted above, it is unclear whether they would be combined into a single .gif file or could remain as separate .gif files. The clerk would then have to define an image set for the .back.gif workspace comprising the plurality of .back.gif files (as a single file or multiple files). Finally, he would have to create an image set for the third workspace comprising one or multiple presumably blank files of the new database system. These individual images that correspond to each other in the first two "image sets" would then somehow have to be related to each other so that all of the corresponding images in the first two workspaces will track each other. That is, when john.doe.gif is in the first workspace, john.doe.back.gif needs to be in the second workspace. How this would be accomplished is not even really discussed in Mishra. Then, the clerk will open the particular session type just created in which the clerk can enter the data from the first two windows into the third window continuously for a plurality of bank customers.

The above-described example makes it quite clear that Mishra discloses a very different technique from the present invention. The workspaces are defined using a very different system and the files that are displayed in them are opened and placed in those workspaces using an entirely different technique.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: 9.17.03



Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: 215-923-4466
Facsimile: 215-923-2189

Attorney for Applicants
International Business
Machines Corporation